SELF-DISTILLED POLICY GRADIENT

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation, where a language model conditions on privileged context to supervise its own generations, is a promising source of dense supervision for sparse-reward reinforcement learning. We instantiate OPD as an auxiliary full-vocabulary student-to-teacher reverse Kullback–Leibler divergence (KL) loss. For any fixed sampled prefix and with the privileged branch detached, we show that its student-side gradient is identical to a policy-gradient update whose centered token advantage is a centered log teacher/student ratio. We propose SDPG, a self-distilled policy-gradient framework that combines standard-deviation-normalized group-relative verifier advantages, exact full-vocabulary OPD, and reference-policy KL regularizationing. Positive-advantage gating and a warmup-decay schedule control when the privileged signal is trusted. Empirically, SDPG improves stability and performance over RLVR and self-distillation baselines.

1 INTRODUCTION

With the development of Reinforcement Learning with Verifiable Rewards (RLVR), Large Language Models (LLMs) have demonstrated remarkable capabilities in complex reasoning tasks such as mathematics and code generation. Algorithms in this family, such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024), optimize against rule-based outcome rewards and have become the standard recipe for post-training reasoning models, eliminating the cost and bias of human preference annotation.

Despite its success, RLVR encounters several limitations, including sparse sequence-level reward across tokens and instability under negative advantages during the early stages of training. Although recent works such as Dr.GRPO (Liu et al., 2025), DAPO (Yu et al., 2025), and GSPO (Zheng et al., 2025) address the latter through asymmetric dual-clip thresholds and sequence-level advantage, the sparsity issue remains unresolved.

Recently, on-policy distillation (OPD) approaches have been proposed to yield dense token-level signals (Agarwal et al., 2024; Lu & Lab, 2025; Fu et al., 2026). Such methods maintain two models: a student model to be optimized that rolls out trajectories, and a teacher model that produces token-level guidance via Kullback–Leibler divergence (KL) regularization or related objectives (Gu et al., 2024; Xu et al., 2025; Yang et al., 2025). However, traditional distillation approaches use a much larger and stronger teacher, which imposes a considerable memory burden when optimizing student models. Moreover, heterogeneous teacher signals may hurt the smoothness of the training process.

A recent line of work addresses such limitations through on-policy self-distillation. In these methods, the teacher model is exactly the same model as the student model, but with additional knowledge such as demonstrations, direct answers, and reasoning paths (Hübotter et al., 2026; Shenfeld et al., 2026; Penaloza et al., 2026). This converts sparse and inconsistent outcome rewards into dense, per-token, and homogeneous supervision. In detail, OPCD (Ye et al., 2026) involves in-context knowledge in the teacher model and internalizes it into the student model through KL divergence; OPSD (Zhao et al., 2026) applies a full, vocabulary-wise KL divergence for better reasoning performance; and TRRD (Zhang et al., 2026b) incorporates trust regions in distillation.

However, the phrase "self-distillation" can obscure a useful policy-gradient interpretation. For a fixed rollout prefix $(x, y_{<t})$, let

$$q_t(a) = \pi_\theta(a \mid c, x, y_{<t}) \quad \text{and} \quad p_t(a) = \pi_\theta(a \mid x, y_{<t}),$$

where q_t is the privileged distribution induced by the same model under context c, and p_t is the deployable distribution without privileged context. We use the full-vocabulary reverse KL $D_{\mathrm{KL}}(p_t \| q_t)$

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

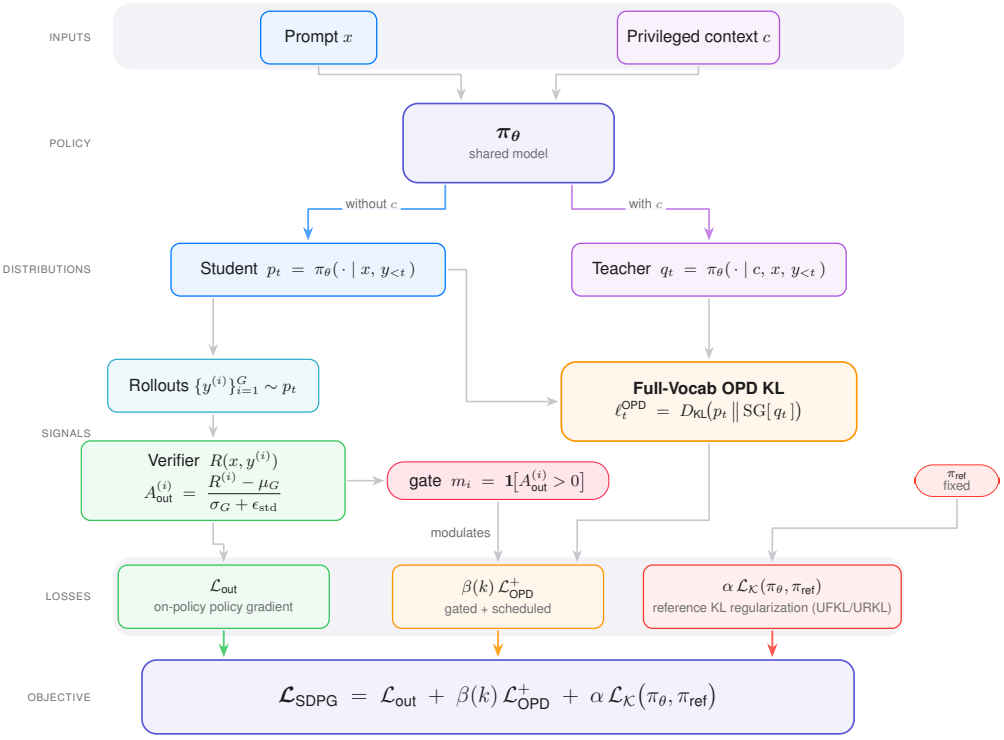


Figure 1: Overview of the Self-Distilled Policy Gradient (SDPG) objective, combining rollout-based outcome optimization, gated On-Policy Distillation (OPD) from privileged context, and a KL regularization to a fixed reference policy.

between these two distributions. Conditioned on the sampled prefix and with the privileged branch detached, its student-side gradient is locally identical to a detached-sampling policy-gradient update whose token advantage is a centered log teacher/student ratio. This is a gradient identity, not a replacement of the full-vocabulary KL objective. It also suggests a failure mode of pure self-distillation: without a verifier, a privileged but imperfect distribution can reinforce locally plausible tokens on globally wrong trajectories. A natural remedy is to combine the advantages of both self-distillation and RLVR methods by keeping the GRPO reward objective with self-distillation signals. RLSD (Yang et al., 2026a) incorporates self-distillation in the clipped importance ratio of GRPO loss function to moderate the teacher signal. Nevertheless, (Xiao et al., 2026) still notices that strong teacher signals may hurt the potential of the student models.

In order to address the above challenge, we propose **SDPG** (Self-Distilled Policy Gradient), which integrates exact full-vocabulary privileged OPD into KL-regularized policy optimization. The resulting objective has two complementary sources of supervision: a sparse binary outcome signal from the verifier and a dense full-vocabulary distillation signal from the context-conditioned teacher. Based on the RPG framework (Zhang et al., 2026a), we focus on unnormalized reference-policy KL regularizations in the main text, with normalized variants and all anchor derivations deferred to the appendix. To control noise in the privileged teacher signal, we apply positive-advantage gating and a warmup-decay schedule to the distillation coefficient. Our contributions are:

1. We derive an exact local policy-gradient form of reverse-KL full-vocabulary OPD: for privileged teacher q_t and deployable student p_t, the student-side gradient of $D_{\text{KL}}(p_t \| q_t)$ equals a detached-sampling update with centered log-ratio advantage $(D_{\text{KL}}(p_t \| q_t) - \log(p_t(a)/q_t(a)))$.

2. We propose a KL-regularized policy optimization objective that combines binary outcome rewards with exact full-vocabulary privileged distillation, and instantiate it with rollout-policy-sampled unnormalized forward and reverse KL regularizations to a fixed reference policy.

3. We develop two stabilizers, positive-advantage gating and a warmup-decay schedule for the distillation coefficient, and show that SDPG improves over GRPO and self-distillation baselines.

2 BACKGROUND

Before introducing the SDPG framework, we briefly review GRPO, one of the standard paradigms in RLVR and introduce on-policy self-distillation. After that, we formally define the divergence measures, including the unnormalized KL divergence, which serves as the theoretical foundation for our main KL regularizations.

2.1 GROUP RELATIVE POLICY OPTIMIZATION

In reasoning tasks, such as mathematics or coding, models are typically trained using verifiable outcome-based rewards. Given a prompt x, the model generates a sequence of tokens $y = \{y_1, y_2, \ldots, y_{|y|}\}$. A rule-based verifier assigns a scalar reward $R(x, y)$, e.g., 1 for correct and 0 for incorrect.

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) is the current standard for RLVR. For each prompt x, GRPO samples a group of G outputs $\{y^{(1)}, \ldots, y^{(G)}\}$ from a frozen rollout policy π_{old}, usually the current policy snapshot at the beginning of the update. Instead of training a separate value network, GRPO computes a sequence-level advantage $A^{(i)}$ by normalizing the rewards within the group:

$$A^{(i)} = \frac{R(x, y^{(i)}) - \mu_G}{\sigma_G + \varepsilon_{\text{std}}}, \tag{2.1}$$

where μ_G and σ_G are the mean and standard deviation of the group's rewards, and $\varepsilon_{\text{std}} > 0$ avoids division by zero. Equivalently, implementations often set $A^{(i)} = 0$ when all rewards in the group are identical. The policy is then optimized using a PPO-style (Schulman et al., 2017) clipped surrogate objective:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\mathbb{E}_{x,\, \{y^{(i)}\}_{i=1}^{G} \sim \pi_{\text{old}}(\cdot|x)} \left[\frac{1}{\sum_i |y^{(i)}|} \sum_{i=1}^{G} \sum_{t=1}^{|y^{(i)}|} \min\Big(r_{i,t} A^{(i)},\ \text{clip}(r_{i,t}, 1-\epsilon, 1+\epsilon) A^{(i)} \Big) \right], \tag{2.2}$$

where $r_{i,t}$ is the importance ratio defined as follows:

$$r_{i,t} = \frac{\pi_\theta(y_t^{(i)} \mid x, y_{<t}^{(i)})}{\pi_{\text{old}}(y_t^{(i)} \mid x, y_{<t}^{(i)})}$$

is the importance sampling ratio. During the policy update, π_{old} is fixed and gradients are taken only through π_θ.

While GRPO is computationally efficient, the scalar advantage $A^{(i)}$ is applied equally to every token in the sequence. This sparse, sequence-level credit assignment makes it difficult for the model to identify which specific reasoning steps were correct or flawed, leading to inefficient exploration. Furthermore, standard PPO clipping struggles with the overwhelming number of negative advantages ($A < 0$) early in training, which can degrade the model's foundational language capabilities.

2.2 ON-POLICY SELF-DISTILLATION

In standard off-policy knowledge distillation, a smaller student model is trained to mimic the behavior of a more capable, distinct teacher model. The conventional objective minimizes the divergence between the teacher's and student's output distributions over trajectories generated by the teacher. While effective, this off-policy paradigm suffers from exposure bias and distribution mismatch, as the student is trained on the teacher's distribution but evaluated on its own rollouts during inference.

On-policy self-distillation reduces distribution mismatch by evaluating the distillation loss on prefixes sampled from the student's rollout distribution, usually a frozen snapshot of the current policy. It also eliminates the need for an external teacher model: a single model π_θ acts as the deployable student when conditioned on x and as the privileged teacher when additionally conditioned on c, such as ground-truth reasoning traces or verified reference answers. At prefix $(x, y_{<t})$, the two distributions are

$$q_t(a) = \pi_\theta(a \mid c, x, y_{<t}), \qquad p_t(a) = \pi_\theta(a \mid x, y_{<t}). \tag{2.3}$$

Prior work usually aligns these distributions through a full-vocabulary KL divergence. In this work we use the student-to-teacher reverse KL $D_{\mathrm{KL}}(p_t \| q_t)$. Proposition A.1 shows that, on a fixed sampled prefix and with the teacher branch detached, its student-side gradient has an equivalent local policy-gradient form.

2.3 Unnormalized KL Divergence

The standard Normalized Kullback-Leibler (KL) divergence between two distributions P and Q is defined as:

$$D_{\mathrm{KL}}(P\|Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx.$$

However, in the scenario when reference measures may not be perfectly normalized (Zhang et al., 2026a) (i.e., $\int_x P(x)$ and $\int_x Q(x)$ may be not equal to 1), to yield a more elegant and symmetric gradient, we employ the Unnormalized KL (UKL) divergence (Zhu & Rohwer, 1995; Minka, 2005). UKL introduces a mass correction term to handle scenarios where the distributions sum over sub-vocabularies or are otherwise unnormalized. It is defined as:

$$\mathrm{UKL}(P\|Q) = \underbrace{\int P(x) \log \frac{P(x)}{Q(x)} dx}_{D_{\mathrm{KL}}(P\|Q)} + \underbrace{\int \big(Q(x) - P(x)\big) dx}_{\text{Mass Correction}}. \tag{2.4}$$

Actually, it is equivalent to k_3 estimator (Schulman, 2020), which is an unbiased estimator of KL divergence but with lower variance. While we will employ the exact token-level KL over the full vocabulary for the self-distillation objective, we will use this unnormalized KL (UKL) framework for the policy regularization against the reference model π_{ref}.

3 Self-Distilled Policy Gradient

SDPG has three components. First, it uses on-policy objective as used in RLVR, with rewards supplied by a binary verifier. Second, it adds exact full-vocabulary OPD on prefixes sampled from the unprivileged rollout policy, so the privileged signal is dense but remains tied to the student's own state distribution. Third, it anchors the updated policy to a fixed reference policy through a rollout-policy-sampled KL surrogate. Algorithm 1 summarizes the training loop.

3.1 KL-Regularized Policy Optimization with Outcome and OPD

SDPG minimizes a KL-regularized policy optimization objective with two sources of supervision:

$$\mathcal{L}_{\mathrm{SDPG}}(\theta) = \mathcal{L}_{\mathrm{out}}(\theta) + \beta(k)\mathcal{L}^{+}_{\mathrm{OPD}}(\theta) + \alpha\mathcal{L}_{\mathcal{K}}(\pi_\theta, \pi_{\mathrm{ref}}), \tag{3.1}$$

where $\mathcal{L}_{\mathrm{out}}$ is the reward-based policy-gradient loss, $\mathcal{L}^{+}_{\mathrm{OPD}}$ is the gated full-vocabulary OPD loss, $\beta(k)$ is the distillation coefficient at training step k, and $\mathcal{L}_{\mathcal{K}}$ is a KL regularization against the fixed reference policy. When $\beta = 0$, SDPG reduces to the corresponding RPG-style objective. However, the forms of $\mathcal{K}$ for the on-policy loss are quite different from Zhang et al. (2026a), which is under a one-step off-policy setting. When $\alpha = 0$, SDPG becomes outcome-reward policy optimization with full-vocabulary OPD but without a reference-policy anchor.

3.2 On-policy Reward-based Loss

We first derive the reward-based loss $\mathcal{L}_{\mathrm{out}}$ in Eq. 3.1. For full on-policy specialization, the rollout policy is exactly the current policy π_θ. Therefore, $y_i \sim \mathrm{SG}[\pi_\theta(\cdot \mid x)]$, where SG is the stop-gradient operator. For the objective

$$J_{\mathrm{out}}(\theta) = \mathbb{E}_{x,y\sim\pi_\theta(\cdot|x)}[R(y)],$$

to generate gradient signals from reward, we should change to REINFORCE-style surrogate loss Williams (1992):

$$\mathcal{L}_{\mathrm{out}}(x_i, y_i, \theta) = -R(y_i) \log \pi_\theta(x_i) \tag{3.2}$$

Algorithm 1 SDPG: Self-Distilled Policy Gradient with Full-Vocabulary OPD

Require: Training data $\mathcal{D} = \{(x, c)\}$, where x is the input and c is privileged in-context knowledge; language model π_θ; fixed reference policy π_{ref}; total training steps T; $\epsilon_{\text{std}} > 0$.
for each training step $k = 1, \ldots, T$ **do**
  Sample a batch of prompts and privileged contexts $\{(x_j, c_j)\}_{j=1}^{B} \sim \mathcal{D}$
  for each prompt x_j **do**
    // Rollout from the frozen unprivileged behavior policy
    Sample a group of G responses $\{y_j^{(i)}\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid x_j)$
    // Compute outcome rewards and group-relative advantages
    Obtain binary verifier rewards $R_j^{(i)} = R(x_j, y_j^{(i)})$
    Compute $A_{\text{out}}^{(i)} = \frac{R_j^{(i)} - \mu_j}{\sigma_j + \epsilon_{\text{std}}}$, where μ_j and σ_j are the mean and standard deviation of $\{R_j^{(i)}\}_{i=1}^{G}$
    // The positive advantage gating
    Set $m_j^{(i)} = \mathbf{1}[A_{\text{out}}^{(i)} > 0]$
    for each response $y_j^{(i)}$ and token position t **do**
      Define the prefix state $s_{j,i,t} = (x_j, y_{j,<t}^{(i)})$
      // Student, privileged-teacher, behavior, and reference distributions
      Compute $p_{j,i,t} = \pi_\theta(\cdot \mid s_{j,i,t})$, $q_{j,i,t} = \text{SG}[\pi_\theta(\cdot \mid c_j, s_{j,i,t})]$ and $\pi_{\text{ref}}(\cdot \mid s_{j,i,t})$
      // Exact full-vocabulary OPD loss on this sampled prefix
      $\mathcal{L}_{j,i,t}^{\text{OPD}} = \sum_{a \in \mathcal{V}} p_{j,i,t}(a) \log \frac{p_{j,i,t}(a)}{q_{j,i,t}(a)}$
    end for
  end for
  Update θ by minimizing $\mathcal{L}_{\text{SDPG}}(\theta)$ in Eq. 3.1
end for
return π_θ

so that $\mathbb{E}_{x \sim \pi_\theta}[\nabla_\theta \mathcal{L}_{\text{out}}(x, y, \theta)] = -\nabla_\theta J_{\text{out}}(\theta)$. Therefore, no PPO-style importance-ratio clipping is needed. Moreover, similar as GRPO (Shao et al., 2024), to mitigate the variance and neutralize baseline bias, we use the same group-relative advantage $A_{\text{out}}(y)$ as in Eq. 2.1 instead of the original reward $R(y)$. Therefore, the verifier-grounded objective can be written directly as

$$\mathcal{L}_{\text{out}}(\theta) = -\mathbb{E}_{(x,c)\sim\mathcal{D},\ \{y_i\}_{i=1}^{G}\sim\pi_\theta(\cdot|x)}\left[\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|} A_{\text{out}}(y_i)\ \log \pi_\theta(y_{i,t} \mid x, y_{i,<t})\right]. \quad (3.3)$$

3.3 Full-Vocabulary Distillation on Sampled Prefixes

Instead of using a sampled-token approximation to the privileged teacher signal, SDPG uses the exact full-vocabulary student-to-teacher KL on each sampled prefix similar to Zhao et al. (2026). For a sampled response y_i and token position t, define the prefix $s_{i,t} = (x, y_{i,<t})$ and the two next-token distributions

$$p_{i,t}(a) = \pi_\theta(a \mid x, y_{i,<t}),$$
$$q_{i,t}(a) = \pi_\theta(a \mid c, x, y_{i,<t}).$$

With the sampled prefix and teacher branch detached, the per-token OPD loss is

$$\ell_{i,t}^{\text{OPD}}(\theta) = D_{\text{KL}}(p_{i,t} \| \text{SG}[q_{i,t}]) = \sum_{a \in \mathcal{V}} p_{i,t}(a) \log \frac{p_{i,t}(a)}{\text{SG}[q_{i,t}(a)]}. \quad (3.4)$$

The total distillation loss over sampled sequences is

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{(x,c)\sim\mathcal{D},\ \{y_i\}_{i=1}^{G}\sim\pi_\theta(\cdot|x)}\left[\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\ell_{i,t}^{\mathrm{OPD}}(\theta)\right]. \tag{3.5}$$

3.4 On-policy Unnormalized KL for SDPG

Now we focus on the KL regularization term $\mathcal{L}_{\mathcal{K}}$ in Eq. 3.1. Due to page limitations, we postpone the derivations for general forward and reverse KL regularization to Appendix C.1. From the derivation there, it is worth noting that $\pi_{\mathrm{ref}} = \pi_\theta$ is not sufficient to minimize the surrogate loss for forward and reserve KL, which is due to the inherent biases within (normalized) forward and backward KL regularization. To tackle this mismatching issue, we apply the unnormalized KL term as introduced in Section 2.3. For simplicity, we denote $J_{\mathrm{R\&D}} = J_{\mathrm{out}} + \beta(k)J_{\mathrm{OPD}}$ and $\mathcal{L}_{\mathrm{R\&D}} = \mathcal{L}_{\mathrm{out}} + \beta(k)\mathcal{L}_{\mathrm{OPD}}$ for training step k as the objective and loss functions for reward-based and distillation terms, respectively.

In detail, consider the objective using unnormalized forward KL regularization as follows:

$$J_{\mathrm{SDPG-UFKL}}(\theta) = J_{\mathrm{R\&D}}(\theta) - \alpha\mathrm{UKL}(\pi_{\mathrm{ref}}\|\pi_\theta),$$

where J_{OPD} is the objective of on-policy distillation implicitly involved in Eq. 3.5. The gradient, expressed as an expectation over π_θ using $w_T(x) = \pi_\theta(x)/\pi_{\mathrm{teacher}}(x)$, $w_R = \pi_\theta(x)/\pi_{\mathrm{ref}}(x)$, $\pi_{\mathrm{teacher}}(x) = \pi_\theta(x, c)$ is:

$$\nabla_\theta J_{\mathrm{SDPG-UFKL}}(\theta) = \nabla_\theta J_{\mathrm{R\&D}}(\theta) - \alpha\,\mathbb{E}_{x\sim\pi_\theta}\left[(1 - w_R(x)^{-1})\nabla_\theta \log \pi_\theta(x)\right].$$

A corresponding differentiable surrogate loss term for minimization via gradient descent is (ignoring the prefix $y_{<t}$ already generated):

$$\mathcal{L}_{\mathrm{SDPG-UFKL}}(x,\theta) = \mathcal{L}_{\mathrm{R\&D}}(x,\theta) + \alpha(w_R(x)^{-1} + \log w_R(x)), \tag{3.6}$$

such that $\mathbb{E}_{x\sim\pi_\theta}[\nabla_\theta\mathcal{L}_{\mathrm{SDPG-UFKL}}(x,\theta)] = -\nabla_\theta J_{\mathrm{SDPG-UFKL}}(\theta)$. Therefore, the total surrogate loss function can be written as

$$\mathcal{L}_{\mathrm{SDPG-UFKL}}(\theta) = \mathcal{L}_{\mathrm{R\&D}}(\theta) + \alpha\,\mathbb{E}_{(x,c)\sim\mathcal{D},y\sim\pi_\theta(\cdot|x)}\left[\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\left(\frac{\pi_{\mathrm{ref}}(y_{i,t}\mid x, y_{i,<t})}{\pi_\theta(y_{i,t}\mid x, y_{i,<t})} + \log\frac{\pi_\theta(y_{i,t}\mid x, y_{i,<t})}{\pi_{\mathrm{ref}}(y_{i,t}\mid x, y_{i,<t})}\right)\right].$$

Moreover, we can also apply the unnormalized reverse KL regularization as follows:

$$J_{\mathrm{SDPG-URKL}}(\theta) = J_{\mathrm{R\&D}}(\theta) - \alpha\mathrm{UKL}(\pi_\theta\|\pi_{\mathrm{ref}}).$$

The gradient, expressed as an expectation over π_θ is:

$$\nabla_\theta J_{\mathrm{SDPG-URKL}}(\theta) = \nabla_\theta J_{\mathrm{R\&D}}(\theta) - \alpha\,\mathbb{E}_{x\sim\pi_\theta}\left[(\log w_R(x))\nabla_\theta \log \pi_\theta(x)\right].$$

A corresponding differentiable surrogate loss term for minimization via gradient descent is (ignoring the prefix $y_{<t}$ already generated):

$$\mathcal{L}_{\mathrm{SDPG-URKL}}(x,\theta) = \mathcal{L}_{\mathrm{R\&D}}(x,\theta) + \frac{\alpha}{2}\log^2 w_R(x), \tag{3.7}$$

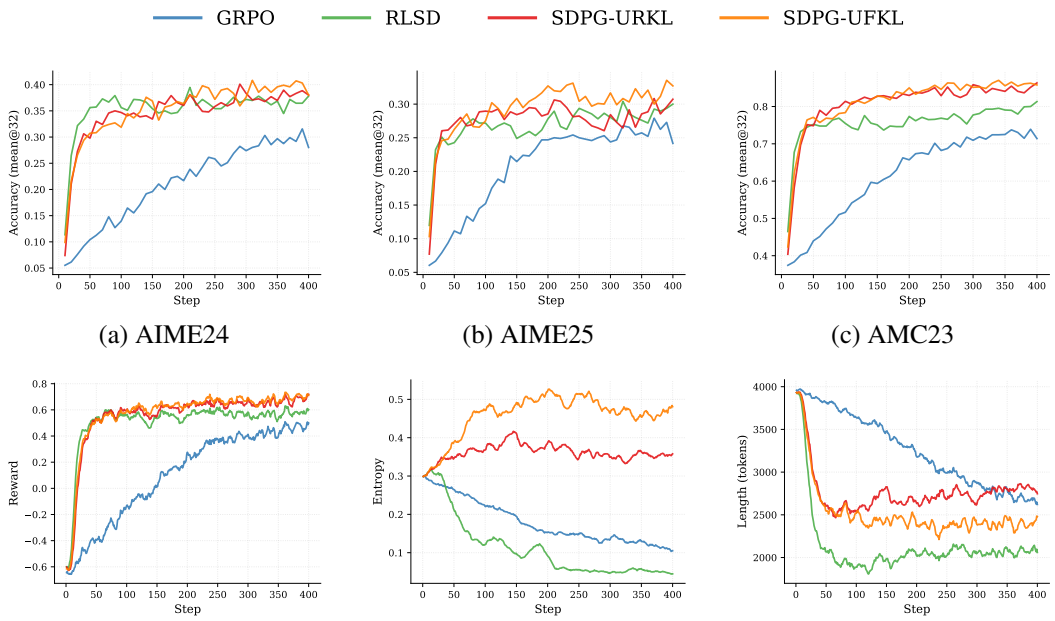
such that $\mathbb{E}_{x\sim\pi_\theta}[\nabla_\theta\mathcal{L}_{\mathrm{SDPG-URKL}}(x,\theta)] = -\nabla_\theta J_{\mathrm{SDPG-URKL}}(\theta)$. And the total surrogate loss function can be written as

$$\mathcal{L}_{\mathrm{SDPG-URKL}}(\theta) = \mathcal{L}_{\mathrm{R\&D}}(\theta) + \alpha\,\mathbb{E}_{(x,c)\sim\mathcal{D},y\sim\pi_\theta(\cdot|x)}\left[\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\frac{1}{2}\log^2\frac{\pi_\theta(y_{i,t}\mid x, y_{i,<t})}{\pi_{\mathrm{ref}}(y_{i,t}\mid x, y_{i,<t})}\right].$$

3.5 Additional Stabilizers for SDPG

SDPG uses two lightweight controls to prevent the privileged OPD signal from overwhelming verifier-grounded learning. First, OPD is applied only to trajectories with positive group-relative outcome advantage:

$$m_i = \mathbf{1}[A_{\mathrm{out}}^{(i)} > 0], \qquad \mathcal{L}_{\mathrm{OPD}}^{+} = \mathbb{E}\left[\frac{1}{\sum_i|y_i|}\sum_{i,t} m_i\ell_{i,t}^{\mathrm{OPD}}\right].$$



(a) AIME24 (b) AIME25 (c) AMC23

(d) Reward (e) Entropy (f) Response Length

Figure 2: Training dynamics and benchmark performance on Qwen3-4B trained with baseline algorithms and SDPG variants. **Top row:** (a) AIME24, (b) AIME25, (c) AMC23. **Bottom row:** (d) group-relative reward, (e) actor entropy, (f) average response length. Both SDPG-URKL and SDPG-UFKL reach higher final accuracies than baselines. SDPG-UFKL also avoids the entropy collapse observed in RLSD.

Second, the OPD coefficient is warmed up and then decayed:

$$\beta(k) = \beta_{\text{base}} \min\left(1, \frac{k}{T_{\text{warm}}}\right) \min\left(1, \frac{T-k}{T_{\text{decay}}}\right).$$

These choices trust privileged distillation only on verifier-endorsed rollouts and phase out the privileged signal near the end of training. Detailed motivation, edge cases, and the schedule visualization are deferred to Appendix A.2.

4 Experiments

In this section, we empirically evaluate our proposed SDPG algorithm and compare the performance against baselines on challenging mathematical reasoning tasks based on pretrained LLMs, including GRPO (Shao et al., 2024) and RLSD (Yang et al., 2026a).

4.1 Experiment Settings

We conduct experiments primarily on Qwen3-4B (Yang et al., 2025); additional results on Qwen3-1.7B with the further baseline, OPCD, are reported in Appendix E.2. For training, we utilize the DAPO-Math-17k dataset (Yu et al., 2025) with 13.9k English samples and generate the privileged information using Gemini 2.5 Pro (Comanici et al., 2025). And the prompts for teacher and student models are shown in Figure 4 in the Appendix. We evaluate the fine-tuned models on the benchmarks of AIME2024 (MAA, 2024a;b), AIME2025 (MAA, 2025a;b), and AMC23 (MAA, 2023). Experiments are implemented using the verl framework (Sheng et al., 2025) with the vLLM engine (Kwon, 2025) for efficient LLM serving and inference.

All experiments use the AdamW optimizer (Loshchilov & Hutter, 2018) with a learning rate of 1×10^{-6}, a weight decay of 0.1 $(\beta_1, \beta_2) = (0.9, 0.999)$, and gradient clipping at 1.0. Training proceeds for 400 steps with 10 warmup steps in the beginning. The global training batch size is 128, with 8 responses per prompt and a temperature of 1.0. We use FSDP with bfloat16 mixed

Table 1: Performance of models trained with baselines and SDPG variants on AIME24, AIME25, and AMC23 (pass@1, mean@32) with Qwen3-4B trained for 400 steps. *Last* shows the score at step 400; *Best* shows the peak across training. Column maximum in **bold**, second-best in underlined.

	AIME24		AIME25		AMC23	
Method	Last	Best	Last	Best	Last	Best
GRPO	0.280	0.316	0.242	0.279	0.714	0.739
RLSD	0.378	0.395	0.300	0.304	0.813	0.813
SDPG-URKL (ours)	0.380	0.401	0.307	0.308	**0.863**	0.863
SDPG-UFKL (ours)	**0.380**	**0.408**	**0.327**	**0.335**	0.858	**0.870**

precision and the vLLM rollout engine, and all experiments are conducted on 8 NVIDIA H100 GPUs. The maximum prompt and response lengths are set to 2,048 and 4,096, respectively, with dynamic batching enabled. For all the baselines, we use $\epsilon_{\text{std}} = 1 \times 10^{-6}$ in Eq. 2.1, and use a clipping threshold of $(\epsilon_1, \epsilon_2) = (0.2, 0.2)$, and the KL regularization coefficient is set to 1×10^{-3}. For SDPG, we use $\alpha = 1 \times 10^{-3}$, $\beta_{\text{base}} = 1 \times 10^{-3}$, $T_{\text{warm}} = 50$, and $T_{\text{decay}} = 350$.

4.2 Experiment Results

The quantitative results in Table 1 demonstrate the competitive performance of the proposed SDPG framework. Both SDPG-URKL and SDPG-UFKL outperform GRPO and RLSD across all the benchmarks, with SDPG-UFKL achieving the top score in five and SDPG-URKL in the remaining one. Figure 2 complements these results by illustrating evaluation scores and training dynamics. The accuracy gap between SDPG and GRPO opens within the first 50 steps and persists throughout training (Figures 2a–c), and SDPG reaches the high-reward plateau (Figure 2d) several hundred steps earlier than GRPO. Notably, SDPG-UFKL maintains substantially higher actor entropy throughout training (Figure 2e), in contrast to RLSD whose entropy collapses toward zero by step 250, a known signature of mode collapse in pure self-distillation. We attribute this stability to the combination of positive-advantage gating and the warmup-decay β schedule, which together prevent privileged distillation from over-constraining the policy after the student has internalized its useful signal. Response lengths (Figure 2f) for SDPG methods stabilize at intermediate values, sufficient for multi-step reasoning while remaining shorter than GRPO's verbose outputs. Ablation studies isolating the contributions of the OPD distillation term and the KL regularization, as well as additional results on Qwen3-1.7B, are reported in Appendix E.

5 Related Work

Reinforcement Learning with Verifiable Rewards. Reinforcement Learning with Verifiable Rewards (RLVR) (Wang et al., 2025b) has become one of the dominant techniques in post-training stages of LLMs, which reward models with an automatic verifier for achieving correct final answers regardless of intermediate reasoning steps (Srivastava & Aggarwal, 2025). It eliminates the need for human preference labels in previous algorithms like PPO (Schulman et al., 2017). One classic paradigm is GRPO (Shao et al., 2024), which utilizes group-relative advantage, PPO-style clipped surrogate, as well as KL regularization to achieve competitive performances when training LLMs. However, GRPO has limitations: advantages collapse and learning stalls when all rollouts in the group have identical rewards, and token-wise credit assignment remains sparse in long generations. Therefore, a lot of improved algorithms have been proposed. Dr. GRPO (Liu et al., 2025) removes biased normalization to improve token efficiency. VinePPO (Kazemnejad et al., 2024) utilizes Monte-Carlo-based value estimation to improve credit assignment. Moreover, DAPO (Yu et al., 2025) introduces techniques including clip-higher, dynamic sampling, token-level loss, and soft overlong punishment to enhance downstream performance, while GSPO (Zheng et al., 2025) uses sequence likelihood in the importance ratio to further stabilize training. RPG (Zhang et al., 2026a) provides a unified framework for different types of KL regularization in GRPO.

However, these approaches may suffer from signal homogeneity among all tokens in a sequence. Some research utilizes process reward models (Lightman et al., 2023; Wang et al., 2024; Chen et al., 2024; Zhang et al., 2025; Dai et al., 2025; Yang et al., 2026b; Luo et al., 2024) or step-level value

estimators for intermediate signals. However, expensive human step annotations may make these methods less affordable. Beyond these, other works try to use per-token signals such as entropy, key-token statistics, uncertainty, or attention dynamics to circumvent expensive annotations (Xie et al., 2025; Li et al., 2026; Cheng et al., 2026; Wang et al., 2025a; Chen et al., 2025). However, such utilization of intrinsic signals may be heuristic. Our method instead incorporates full-vocabulary privileged self-distillation while retaining the binary outcome verifier.

On-Policy Distillation and Self-Distillation. To deal with sparse rewards in reinforcement learning, on-policy distillation (OPD) techniques have been proposed to provide dense token-level supervision signals (Agarwal et al., 2024; Lu & Lab, 2025; Fu et al., 2026). They train a student model based on trajectories sampled from its own policy, while another teacher model provides token-level targets via KL regularization or related objectives (Gu et al., 2024; Xu et al., 2025; Yang et al., 2025; Xiao et al., 2026). However, these OPD methods often require a large external teacher, which requires much larger memory usage and may provide mismatched guidance due to model heterogeneity.

To address such issues, self-distillation methods sample from the same student policy but evaluate the model under privileged knowledge, including ground-truth solution paths and environmental feedback (Hübotter et al., 2026; Shenfeld et al., 2026; Penaloza et al., 2026). In detail, OPSD (Zhao et al., 2026) utilizes full KL divergence between teacher and student models to enhance the reasoning ability of LLMs; OPCD (Ye et al., 2026) adds flexibility by decoupling the on-policy strategy; and TRRD (Zhang et al., 2026b) involves the teacher policy in the importance ratio to alleviate the conflict between reward function and distillation term. However, pure on-policy self-distillation approaches suffer from limited exploration and mode collapse. RLSD (Yang et al., 2026a) instead uses the privileged teacher-student likelihood ratio as a token-level credit reweighting signal inside a GRPO-style objective; we summarize this distinction in Section B and provide the full loss in Appendix B. By taking advantage of RLVR techniques, the training process becomes smoother. However, strong teacher signals can still impose a lower ceiling on the student (Xiao et al., 2026). SDPG differs from these methods by preserving the exact full-vocabulary OPD objective while coupling it with verifier-based policy optimization and reference-policy KL regularizationing.

6 Conclusion

We presented SDPG, a Self-Distilled Policy Gradient framework that combines verifier-based RLVR with exact full-vocabulary OPD. In this view, the reverse-KL OPD term remains a full-vocabulary distillation objective, while its fixed-prefix student-side gradient admits an equivalent policy-gradient form with a centered log-ratio token advantage. Combining this OPD-equivalent distillation signal with binary verifier rewards improves credit assignment while retaining the exploration and selection benefits of RLVR. On LLM reasoning tasks, the proposed algorithms achieve better performance and stability than baseline algorithms.

AI use statement

In this work, we used generative AI tools for generate synthetic data sets. We have not used generative AI tools for help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. Step-level value preference optimization for mathematical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7889–7903, 2024.

Minghan Chen, Guikun Chen, Wenguan Wang, and Yi Yang. Seed-grpo: Semantic entropy enhanced grpo for uncertainty-aware policy optimization. *arXiv preprint arXiv:2505.12346*, 2025.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30377–30385, 2026.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. URL https://arxiv.org/abs/2507.06261.

Mz Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *The twelfth international conference on learning representations*, 2024.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in rl training of llms. *arXiv preprint arXiv:2410.01679*, 2024.

Woosuk Kwon. *vLLM: An Efficient Inference Engine for Large Language Models*. PhD thesis, UC Berkeley, 2025.

Bruce W Lee, Hyunsoo Cho, and Kang Min Yoo. Instruction tuning with human curriculum. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 1281–1309, 2024.

Ziheng Li, Liu Kang, Feng Xiao, Luxi Xing, Qingyi Si, Zhuoran Li, Weikang Gong, Deqing Yang, Yanghua Xiao, and Hongcheng Guo. Outcome-grounded advantage reshaping for fine-grained credit assignment in mathematical reasoning. *arXiv preprint arXiv:2601.07408*, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The twelfth international conference on learning representations*, 2023.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

K. Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. URL https://thinkingmachines.ai.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, et al. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*, 2024.

Mathematical Association of America's American Mathematics Competitions MAA. 2023 AMC, 2023. URL https://artofproblemsolving.com/wiki/index.php/AMC_12_Problems_and_Solutions.

Mathematical Association of America's American Mathematics Competitions MAA. 2024 AIME-I, 2024a. URL https://artofproblemsolving.com/wiki/index.php/2024_AIME_I. Accessed: 2025-05-08.

Mathematical Association of America's American Mathematics Competitions MAA. 2024 AIME-II, 2024b. URL https://artofproblemsolving.com/wiki/index.php/2024_AIME_II. Accessed: 2025-05-08.

Mathematical Association of America's American Mathematics Competitions MAA. 2025 AIME-I, 2025a. URL https://artofproblemsolving.com/wiki/index.php/2025_AIME_I.

Mathematical Association of America's American Mathematics Competitions MAA. 2025 AIME-II, 2025b. URL https://artofproblemsolving.com/wiki/index.php/2025_AIME_II.

Thomas Minka. Divergence measures and message passing. 2005.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. *arXiv preprint arXiv:2602.04942*, 2026.

John Schulman. Approximating kl divergence. http://joschu.net/blog/kl-approx.html, March 2020. Accessed on September 26, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Vedant Shah, Johan Obando-Ceron, Vineet Jain, Brian Bartoldson, Bhavya Kailkhura, Sarthak Mittal, Glen Berseth, Pablo Samuel Castro, Yoshua Bengio, Esmeralda S. Whitammer, Moksh Jain, Siddarth Venkatraman, and Aaron Courville. A comedy of estimators: On kl regularization in rl training of llms. *arXiv preprint arXiv:2512.21852*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. Efficient reinforcement finetuning via adaptive curriculum learning. In *NeurIPS 2025 Workshop on Efficient Reasoning*, 2025.

Saksham Sahai Srivastava and Vaneet Aggarwal. A technical survey of reinforcement learning techniques for large language models. *arXiv preprint arXiv:2507.04136*, 2025.

Yunhao Tang and Rémi Munos. On a few pitfalls in kl divergence gradient estimation for rl. *arXiv preprint arXiv:2506.09477*, 2025.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576, 2021.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Tanglifu Tanglifu, Xiaowei Lv, et al. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 318–327, 2025.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

Can Xie, Ruotong Pan, Xiangyu Wu, Yunfei Zhang, Jiayi Fu, Tingting Gao, and Guorui Zhou. Unlocking exploration in rlvr: Uncertainty-aware advantage shaping for deeper reasoning. *arXiv preprint arXiv:2510.10649*, 2025.

Wenda Xu, Rujun Han, Zifeng Wang, Long Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *The Thirteenth International Conference on Learning Representations*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026a.

Chenxu Yang, Qingyi Si, Mz Dai, Dingyu Yao, Mingyu Zheng, Minghui Chen, Zheng Lin, and Weiping Wang. Test-time prompt intervention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34223–34231, 2026b.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yifan Zhang, Yifeng Liu, Huizhuo Yuan, Yang Yuan, Quanquan Gu, and Andrew C Yao. On the design of kl-regularized policy gradient algorithms for llm reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026a.

Zhaoyang Zhang, Shuli Jiang, Yantao Shen, Yuting Zhang, Dhananjay Ram, Shuo Yang, Zhuowen Tu, Wei Xia, and Stefano Soatto. Reinforcement-aware knowledge distillation for llm reasoning. *arXiv preprint arXiv:2602.22495*, 2026b.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Huaiyu Zhu and Richard Rohwer. Information geometric measurements of generalisation. *Preprint*, 1995.

A More on Self-Distilled Policy Gradient

A.1 OPD is a form of policy gradient

To show that our algorithm is also a form of policy gradient, we provide the following proposition:

Proposition A.1 (Fixed-prefix reverse-KL OPD gradient as a policy gradient)**.** Fix a rollout prefix $s_t = (x, y_{<t})$ and write $p_t(a) = \pi_\theta(a \mid x, y_{<t})$ and $q_t(a) = \pi_\theta(a \mid c, x, y_{<t})$. Let $\bar{p}_t = \mathrm{SG}[p_t]$ and $\bar{q}_t = \mathrm{SG}[q_t]$, and assume $\bar{q}_t(a) > 0$ whenever $\bar{p}_t(a) > 0$. With the teacher branch detached, the reverse-KL full-vocabulary OPD loss

$$\mathcal{L}_{\mathrm{OPD},t}(\theta) = D_{\mathrm{KL}}(p_t \| \bar{q}_t) \tag{A.1}$$

has the same student-side gradient, at the current iterate, as the detached-sampling policy-gradient surrogate

$$\widetilde{\mathcal{L}}^{\mathrm{PG}}_{\mathrm{OPD},t}(\theta) = -\mathbb{E}_{a\sim\bar{p}_t}\left[A^{\mathrm{dist}}_t(a) \log p_t(a)\right], \qquad A^{\mathrm{dist}}_t(a) = \bar{D}_t - \log \frac{\bar{p}_t(a)}{\bar{q}_t(a)}, \tag{A.2}$$

where $\bar{D}_t = D_{\mathrm{KL}}(\bar{p}_t \| \bar{q}_t)$. Moreover, A^{dist}_t is centered under the detached student distribution: $\mathbb{E}_{a\sim\bar{p}_t}[A^{\mathrm{dist}}_t(a)] = 0$.

This proposition not only indicates that OPD is also a form of policy gradient, but also shows that reverse-KL OPD reallocates probability mass according to whether a token's teacher/student log-ratio is above or below its student-weighted average. Proposition A.1 is a gradient identity, not an implementation change: SDPG minimizes the explicit full-vocabulary KL in Eq. 3.4, because this leads to more accurate estimation of the gradient.

Proof. For a rollout prefix $s_t = (x, y_{<t})$, recall the unprivileged student distribution and the privileged teacher distribution:

$$p_t(a) = \pi_\theta(a \mid s_t) = \pi_\theta(a \mid x, y_{<t}), \tag{A.3}$$

$$q_t(a) = \pi_{\mathrm{teacher}}(a \mid s_t) = \pi_\theta(a \mid c, x, y_{<t}). \tag{A.4}$$

All identities here are local to a fixed prefix s_t and to the current iterate used to form the detached quantities. Write

$$\bar{p}_t = \mathrm{SG}[p_t], \qquad \bar{q}_t = \mathrm{SG}[q_t], \qquad \bar{D}_t = D_{\mathrm{KL}}(\bar{p}_t \| \bar{q}_t).$$

The sampled-prefix distribution, the teacher branch, and any policy-gradient coefficients are treated as detached, as in standard surrogate optimization. Reverse-KL full-vocabulary on-policy distillation minimizes

$$\mathcal{L}_{\mathrm{OPD},t}(\theta) = D_{\mathrm{KL}}(p_t \| \bar{q}_t) = \sum_{a\in\mathcal{V}} p_t(a) \log \frac{p_t(a)}{\bar{q}_t(a)}. \tag{A.5}$$

The negative student-side gradient of Eq. A.5 is

$$\begin{aligned} -\nabla_\theta \mathcal{L}_{\mathrm{OPD},t}(\theta) &= -\sum_{a\in\mathcal{V}} p_t(a) \left(\log \frac{p_t(a)}{\bar{q}_t(a)} + 1\right) \nabla_\theta \log p_t(a) \\ &= \mathbb{E}_{a\sim\bar{p}_t}\left[-\frac{p_t(a)}{\bar{p}_t(a)} \left(\log \frac{p_t(a)}{\bar{q}_t(a)} + 1\right) \nabla_\theta \log p_t(a)\right], \end{aligned} \tag{A.6}$$

where gradients are evaluated at the iterate satisfying $\bar{p}_t = p_t$. At the iterate where $\bar{p}_t = p_t$,

$$\mathbb{E}_{a\sim\bar{p}_t}[\nabla_\theta \log p_t(a)] = \sum_{a\in\mathcal{V}} p_t(a) \nabla_\theta \log p_t(a) = \nabla_\theta \sum_{a\in\mathcal{V}} p_t(a) = 0.$$

Adding the state-dependent baseline $1 + \bar{D}_t$ therefore leaves the gradient unchanged and yields

$$-\nabla_\theta \mathcal{L}_{\mathrm{OPD},t}(\theta) = \mathbb{E}_{a\sim\bar{p}_t}\left[\left(\bar{D}_t - \log\frac{\bar{p}_t(a)}{\bar{q}_t(a)}\right)\nabla_\theta \log p_t(a)\right], \tag{A.7}$$

which is the negative gradient of the detached-sampling surrogate in Eq. A.2. Centering follows from

$$\mathbb{E}_{a\sim\bar{p}_t}[A_t^{\mathrm{dist}}(a)] = \bar{D}_t - \sum_{a\in\mathcal{V}} \bar{p}_t(a)\log\frac{\bar{p}_t(a)}{\bar{q}_t(a)} = 0.$$

□

Thus, on a fixed sampled prefix and with the teacher branch detached, reverse-KL full-vocabulary OPD has a local policy-gradient interpretation with centered log-ratio advantage. SDPG nevertheless implements the explicit full-vocabulary KL in Eq. A.5; the policy-gradient form is an interpretation of its gradient, not a sampled-token replacement. The same identity also yields sampled-token Monte Carlo estimators by replacing the full-vocabulary expectation with samples from the detached student distribution.

A.2 Additional Stabilizers for SDPG

Because the current model induces the privileged OPD target under a richer context, it can be noisy early in training and over-constrained late in training. We therefore add two simple controls on when and how strongly this distillation signal is used.

A.2.1 Positive Advantage Gating

When a rollout is incorrect ($A_{\mathrm{out}}^{(i)} < 0$), the privileged teacher can still assign high probability to locally plausible tokens on the sampled wrong prefix. Applying full-vocabulary OPD on such prefixes may conflict with the verifier signal: the outcome objective suppresses the trajectory, whereas the distillation objective can still imitate the privileged teacher around that prefix.

We therefore gate the OPD loss by the outcome advantage:

$$m_i = \mathbf{1}[A_{\mathrm{out}}^{(i)} > 0], \qquad \mathcal{L}_{\mathrm{OPD}}^{+}(\theta) = \mathbb{E}_{(x,c)\sim\mathcal{D},\ \{y_i\}_{i=1}^{G}\sim\pi_\theta(\cdot|x)}\left[\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|} m_i \ell_{i,t}^{\mathrm{OPD}}\right]. \tag{A.8}$$

This relies on the full-vocabulary OPD signal only for trajectories that the verifier endorses within the group. If all rewards in a group are identical, both the mean-centered outcome advantage and the OPD gate vanish, avoiding unvalidated distillation on uninformative groups. In the initial stage, the gate may often be inactive, and the binary outcome reward dominates. A training dataset with moderate difficulty or curriculum learning (Wang et al., 2021; Lee et al., 2024; Wen et al., 2025; Shi et al., 2025) is therefore useful for activating the distillation signal. If $m_i = 1$ for all responses, Eq. A.8 reduces to standard full-vocabulary OPD on all sampled prefixes.

A.2.2 β Scheduler

Early misalignment between p_t and the privileged distribution q_t can make the OPD target noisy. To prevent privileged distillation from destabilizing exploration, we warm up β. The OPD term then takes effect gradually after the outcome policy has begun to find correct trajectories.

Moreover, under an idealized privileged-information model, distilling a teacher conditioned on information unavailable to the deployable student can leave an irreducible conditional mutual-information gap, e.g., $I(Y_t; C \mid X, Y_{<t}) > 0$ when C denotes the privileged variable (Yang et al., 2026a). Under our formulation, this means the privileged OPD target may remain biased by information unavailable at inference. Therefore, to release the student and encourage exploration, we decay β at the end of training, phasing out the distillation signal after the student has internalized its useful information.

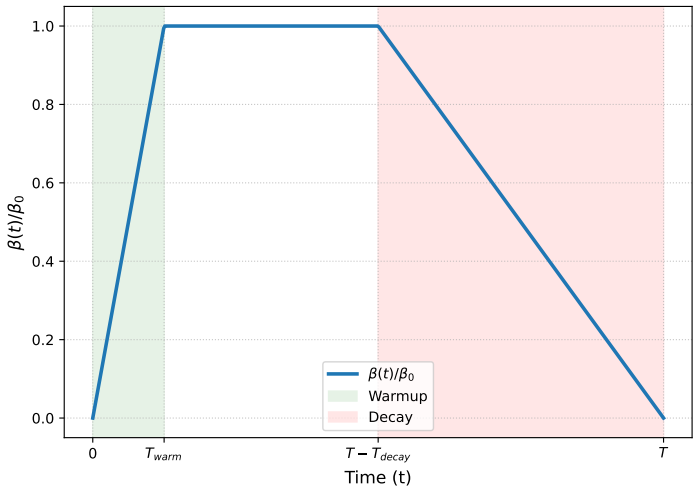


Figure 3: The illustration of the β schedule.

The effective distillation coefficient follows a warmup-decay schedule, illustrated in Figure 3:

$$\beta(k) = \beta_{\text{base}} \times \underbrace{\min\left(1, \frac{k}{T_{\text{warm}}}\right)}_{\text{warmup}} \times \underbrace{\min\left(1, \frac{T-k}{T_{\text{decay}}}\right)}_{\text{decay}}, \tag{A.9}$$

where T_{warm} and T_{decay} are the warmup and decay step counts, and T is the total number of training steps. If the warmup and decay windows overlap, the maximum coefficient can be below β_{base}.

A.3 Discussion about OPSD-style distillation

While we use OPSD-style full-vocabulary on-policy distillation term, deploying this objective in modern distributed training frameworks (e.g., FSDP with vLLM rollouts) requires specific implementation. In this section, we provide a rigorous analysis of the exact loss computed in our implementation, analyzing the discrepancies between the theoretical gradients and the practical approximations.

The full OPSD-style loss function is given by:

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{\substack{(x,c)\sim\mathcal{D} \\ \{y_i\}_{i=1}^{G}\sim\pi_\theta(\cdot|x)}} \left[\frac{1}{\sum_i |y_i|} \sum_{i,t} \mathbb{E}_{v\sim\pi_\theta(\cdot|x,y_{i,<t})} \left[\log \frac{\pi_\theta(v \mid x, y_{i,<t})}{\text{SG}[\pi_\theta(v \mid c, x, y_{i,<t})]} \right] \right].$$

And the corresponding objective is:

$$J(\theta) = \mathbb{E}_{y\sim\pi_\theta} \left[\sum_t D_{\text{KL}}(\pi_\theta \| \text{SG}[\pi_{\text{teacher}}]) \right],$$

where $\pi_{\text{teacher}} = \pi_\theta(\cdot|c,\cdot)$. When differentiating this objective, it yields two distinct gradient paths for the distillation term:

$$\nabla_\theta \mathbb{E}_{y\sim\pi_\theta} \left[\sum_t D_{\text{KL}} \right] = \underbrace{\mathbb{E}_y \left[\nabla_\theta \sum_t D_{\text{KL}} \right]}_{\text{(1) Direct Path-wise Gradient}} + \underbrace{\mathbb{E}_y \left[\sum_t D_{\text{KL}} \cdot \nabla_\theta \log \pi_\theta(y_t) \right]}_{\text{(2) Score Function Gradient}}$$

The direct path-wise gradient term corresponds to $\mathcal{L}_{\text{OPD}}(\theta)$, which is the actual implementation in most modern RL frameworks. However, the score function term is usually omitted, which actually treats the computed KL divergence as an additional negative reward signal. It is worth noting that

the magnitude of this term ($\beta \cdot D_{\mathrm{KL}}$) is negligible compared to the primary sequence-level outcome advantage A_i. And the variance introduced by estimating it would outweigh its marginal theoretical benefit. Therefore, the distillation term acts almost as a local shaping constraint as in term (1). Therefore, we employ the OPSD-style on-policy self-distillation loss as a suitable approximation.

A.4 Teacher and student prompts

We just display the teacher and student prompts in Figure 4, where the solution is generated by Gemini 2.5 Pro. We filter generated solutions using the correct final answers. This substantially reduces incorrect contexts, although a rationale may still contain flawed intermediate reasoning despite reaching the correct answer. We do not claim that every sentence is guaranteed to be correct. The rationale provides privileged guidance, while each rollout remains judged by the exact outcome verifier.

Student Prompt

Solve the following math problem step by step. Present your final answer inside \boxed{}, for example \boxed{42}.

{question}

Remember to put your final answer inside \boxed{}.

Teacher Prompt

Solve the following math problem step by step. Present your final answer inside \boxed{}, for example \boxed{42}.

{question}

Remember to put your final answer inside \boxed{}.
[TEACHER_CONTEXT_TOKEN]

[Hint] The correct answer is {answer}. A common way to solve this is:{Solution}

[Instruction] If possible, derive the answer {answer} using an alternative, equally rigorous mathematical approach (e.g., algebraic vs geometric, or different substitution). If no alternative exists, articulate the standard approach with exceptional clarity. Do NOT state that you were given the answer or reference.

Figure 4: Prompt templates for the student and teacher models.

B Reinforcement Learning with Self-Distillation

RLSD (Yang et al., 2026a) combines verifier-grounded RLVR with privileged self-distillation, but it uses the privileged model only to redistribute token-level credit rather than to define an auxiliary distribution-matching objective. For a sampled token $y_{i,t}$, RLSD computes a stop-gradient privileged information gain $\Delta_{i,t} = \mathrm{SG}[\log q_{i,t}(y_{i,t}) - \log p_{i,t}(y_{i,t})]$, where $q_{i,t}$ is the privileged teacher distribution and $p_{i,t}$ is the deployable student distribution. This gain is exponentiated with the sign of the sequence-level verifier advantage and used to reweight the GRPO token advantage. Consequently, tokens favored by the privileged context receive larger positive credit on successful trajectories, while the ratio is inverted for negative-advantage trajectories.

The key distinction from SDPG is that RLSD does not optimize a separate full-vocabulary distribution-matching OPD loss. The privileged model only changes the magnitude of the verifier-grounded policy-gradient update, whereas SDPG keeps the exact full-vocabulary reverse-KL OPD objective and combines it with outcome-reward optimization.

For a sampled response $y^{(i)}$ and prefix $(x, y_{i,<t})$, define the deployable student distribution and privileged teacher distribution as

$$p_{i,t}(a) = \pi_\theta(a \mid x, y_{i,<t}), \tag{B.1}$$

$$q_{i,t}(a) = \pi_\theta(a \mid c, x, y_{i,<t}). \tag{B.2}$$

RLSD computes the stop-gradient privileged information gain on the sampled token:

$$\Delta_{i,t} = \mathrm{SG}\left[\log q_{i,t}(y_{i,t}) - \log p_{i,t}(y_{i,t})\right]. \tag{B.3}$$

The gain is converted into a direction-aware evidence weight by using the sign of the sequence-level verifier advantage:

$$u_{i,t} = \exp\left(\mathrm{sign}(A^{(i)})\Delta_{i,t}\right) \tag{B.4}$$

$$= \left(\frac{q_{i,t}(y_{i,t})}{p_{i,t}(y_{i,t})}\right)^{\mathrm{sign}(A^{(i)})}. \tag{B.5}$$

The evidence weight is clipped and interpolated with the original GRPO advantage:

$$\widehat{A}^{\mathrm{RLSD}}_{i,t} = A^{(i)}\left[(1-\lambda_{\mathrm{rlsd}}) + \lambda_{\mathrm{rlsd}}\,\mathrm{clip}(u_{i,t}, 1-\epsilon_w, 1+\epsilon_w)\right], \tag{B.6}$$

where $\lambda_{\mathrm{rlsd}} \in [0,1]$ controls the strength of self-distilled credit redistribution and ϵ_w bounds the per-token credit deviation. Setting $\lambda_{\mathrm{rlsd}} = 0$ recovers the uniform GRPO advantage, while $\lambda_{\mathrm{rlsd}} = 1$ gives the fully reweighted RLSD advantage.

Under the minimization convention used in this paper, the corresponding GRPO-style RLSD surrogate is

$$\mathcal{L}_{\mathrm{RLSD}}(\theta) = -\mathbb{E}_{(x,c)\sim\mathcal{D},\ \{y^{(i)}\}_{i=1}^{G}\sim\pi_{\mathrm{old}}(\cdot|x)}\left[\frac{1}{\sum_{i=1}^{G}|y^{(i)}|}\sum_{i=1}^{G}\sum_{t=1}^{|y^{(i)}|}\min\left(\rho_{i,t}\widehat{A}^{\mathrm{RLSD}}_{i,t}, \mathrm{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\widehat{A}^{\mathrm{RLSD}}_{i,t}\right)\right], \tag{B.7}$$

where

$$\rho_{i,t} = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\mathrm{old}}(y_{i,t} \mid x, y_{i,<t})}. \tag{B.8}$$

Equivalently, RLSD replaces the uniform sequence-level GRPO advantage $A^{(i)}$ in Eq. 2.2 with the token-dependent advantage $\widehat{A}^{\mathrm{RLSD}}_{i,t}$. No separate full-vocabulary OPD KL loss is optimized; the privileged teacher affects only the magnitude of token-level credit and not the sign of the verifier-grounded update.

C Additional Discussion on the SDPG loss

C.1 Normalized KL terms

In common implementations such as GRPO (Shao et al., 2024), a forward or reverse KL term is directly applied to the final loss functions. However, if such KL terms are only rollout-based estimation, they could be biased and here we derive the correct loss functions for these KL regularizations.

Firstly, we consider the objective for forward KL regularization as follows:

$$J_{\mathrm{FKL}} = J_{\mathrm{R\&D}} - \alpha D_{\mathrm{KL}}(\pi_{\mathrm{ref}} \| \pi_\theta).$$

The gradient, expressed as an expectation over π_θ using $w_T(x) = \pi_\theta(x)/\pi_{\mathrm{teacher}}(x)$, $w_R = \pi_\theta(x)/\pi_{\mathrm{ref}}(x)$, $\pi_{\mathrm{teacher}}(x) = \pi_\theta(x, c)$ is:

$$\nabla_\theta J_{\mathrm{FKL}}(\theta) = \nabla_\theta J_{\mathrm{R\&D}}(\theta) + \alpha\,\mathbb{E}_{x\sim\pi_\theta}[w_R(x)^{-1}\nabla_\theta \log \pi_\theta(x)].$$

A corresponding differentiable surrogate loss term for minimization via gradient descent is (ignoring the prefix $y_{<t}$ already generated):

$$\mathcal{L}_{\mathrm{FKL}}(x, \theta) = \mathcal{L}_{\mathrm{R\&D}}(x, \theta) + \alpha w_R(x)^{-1},$$

such that $\mathbb{E}_{x\sim\pi_\theta}[\nabla_\theta \mathcal{L}_{\mathrm{FKL}}(x,\theta)] = -\nabla_\theta J_{\mathrm{FKL}}(\theta)$. And the total surrogate loss function can be written as

$$\mathcal{L}_{\mathrm{FKL}}(\theta) = \mathcal{L}_{\mathrm{R\&D}}(\theta) + \alpha\, \mathbb{E}_{(x,c)\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}\left[\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\frac{\pi_{\mathrm{ref}}(y_{i,t}\mid x, y_{i,<t})}{\pi_\theta(y_{i,t}\mid x, y_{i,<t})}\right]$$

For the reverse KL regularization, consider the objective as follows:

$$J_{\mathrm{RKL}} = J_{\mathrm{R\&D}} - \alpha D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}).$$

The gradient, expressed as an expectation over π_θ using $w_T(x) = \pi_\theta(x)/\pi_{\mathrm{teacher}}(x)$, $w_R = \pi_\theta(x)/\pi_{\mathrm{ref}}(x)$, $\pi_{\mathrm{teacher}}(x) = \pi_\theta(x,c)$ is:

$$\nabla_\theta J_{\mathrm{RKL}}(\theta) = \nabla_\theta J_{\mathrm{R\&D}}(\theta) - \alpha\, \mathbb{E}_{x\sim\pi_\theta}[(\log w_R(x) + 1)\nabla_\theta \log \pi_\theta(x)].$$

A corresponding differentiable surrogate loss term for minimization via gradient descent is (ignoring the prefix $y_{<t}$ already generated):

$$\mathcal{L}_{\mathrm{RKL}}(x,\theta) = \mathcal{L}_{\mathrm{R\&D}}(x,\theta) + \frac{\alpha}{2}(\log w_R(x) + 1)^2,$$

such that $\mathbb{E}_{x\sim\pi_\theta}[\nabla_\theta \mathcal{L}_{\mathrm{RKL}}(x,\theta)] = -\nabla_\theta J_{\mathrm{RKL}}(\theta)$. And the total surrogate loss function can be written as

$$\mathcal{L}_{\mathrm{RKL}}(\theta) = \mathcal{L}_{\mathrm{R\&D}}(\theta) + \alpha\, \mathbb{E}_{(x,c)\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}\left[\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\frac{1}{2}\left(1 + \log\frac{\pi_\theta(y_{i,t}\mid x, y_{i,<t})}{\pi_{\mathrm{ref}}(y_{i,t}\mid x, y_{i,<t})}\right)^2\right].$$

It can be observed that due to expectation over π_θ, which is differentiable, although both gradient and expectation operations are linear-operator, they cannot commute and the gradient of expectation is not the same as the expectation of the gradient. Therefore, for rollout-based KL regularization, the original KL loss forms are actually biased. However, as discussed before, due to the inherent biases within normalized forward and backward KL, we apply unnormalized KL in our algorithm, which is also suggested in Zhang et al. (2026a); Tang & Munos (2025) and Shah et al. (2026).

C.2 Equivalence of unnormalized KL with k_3 estimator

In Section 2.3, we note that the Unnormalized KL is equivalent to k_3 estimator. Actually, for $r = \frac{\pi_{\mathrm{ref}}}{\pi_\theta}$, our UFKL sample term is $r - \log(r)$, whereas the standard k_3 estimator is $r - 1 - \log(r)$. The two differ by the constant 1 and therefore have identical gradients. Both are nonnegative up to this constant convention and are stationary at $r = 1$.

C.3 Analysis in one-step off-policy settings

In the implementation of modern RL frameworks, such as verl (Sheng et al., 2025), the rollout model π_{rollout} is often different from the current model π_θ. For example, there would be some sub-iterations of gradient updates for mini-batches in each step. Therefore, there could be issues of within-step off-policy drift and stale importance weights. Therefore, the loss would be different from the full on-policy implementations, and an importance sampling ratio factor should be applied in the loss function. Based on Eq. 3.1, we have

$$\begin{aligned}
\mathcal{L}_{\mathrm{SDPG}}(\theta) &= \mathcal{L}_{\mathrm{out}}(\theta) + \beta(k)\mathcal{L}_{\mathrm{OPD}}(\theta) + \alpha\mathcal{L}_{\mathcal{K}}(\pi_\theta, \pi_{\mathrm{ref}}) \\
&= \mathbb{E}_{(x,c)\sim\mathcal{D},\, \{y_i\}_{i=1}^{G}\sim\pi_\theta(\cdot|x)}\left[\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|} -\mathrm{SG}[A_{\mathrm{out}}(y_i)]\, \log \pi_\theta(y_{i,t}\mid x, y_{i,<t})\right. \\
&\qquad \left. + \beta\mathcal{L}_{\mathrm{OPD}}(x,\theta) + \alpha f_{\mathcal{K}}(\pi_\theta, \pi_R; x, y_i)\right] \\
&= \mathbb{E}_{(x,c)\sim\mathcal{D},\, \{y_i\}_{i=1}^{G}\sim\pi_{\mathrm{rollout}}(\cdot|x)}\left[\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|} -\mathrm{SG}[\rho_{i,t}A_i]\, \log \pi_\theta(y_{i,t}\mid x, y_{i,<t})\right. \\
&\qquad \left. + \beta\mathrm{SG}[\rho_{i,t}]\mathcal{L}_{\mathrm{OPD}}(x,\theta) + \alpha\mathrm{SG}[\rho_{i,t}] f_{\mathcal{K}}(\pi_\theta, \pi_R; x, y_i)\right],
\end{aligned}$$

where

$$\rho_{i,t} = \rho(y_{i,t} \mid x, y_{i,<t}) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\text{rollout}}(y_{i,t} \mid x, y_{i,<t})}$$

is the importance ratio during sampling and $A_i = A_{\text{out}}(y_i)$. And $f_{\mathcal{K}}$ is the corresponding KL surrogate loss.

For the reward-based term, usually we can apply a PPO-style clip to stabilize the training process. In detail, the clipped advantage with the importance ratio is

$$\ell_{t,i}^{\text{clip}} = \begin{cases} \min\big(\max(-A_i\rho_{t,i},\ -A_i\,\text{clip}(\rho_{t,i}, 1{-}\varepsilon_l, 1{+}\varepsilon_h)),\ -A_i c\big) & A_i < 0 \\ \max(-A_i\rho_{t,i},\ -A_i\,\text{clip}(\rho_{t,i}, 1{-}\varepsilon_l, 1{+}\varepsilon_h)) & A_i \geq 0 \end{cases}, \tag{C.1}$$

where ϵ_l, ϵ_h as well as c are the clipping hyperparameters. Here we can set $\epsilon_h = \epsilon_l$ as in GRPO (Shao et al., 2024) and RLSD (Yang et al., 2026a), or $\epsilon_h > \epsilon_l$ as in DAPO (Yu et al., 2025). Therefore, we achieve the final loss in near on-policy (one-step off-policy) approximation implementation:

$$\mathcal{L}_{\text{SDPG}}^{\text{approx}}(\theta) = \mathbb{E}_{(x,c)\sim\mathcal{D},\ \{y_i\}_{i=1}^{G}\sim\pi_{\text{rollout}}(\cdot|x)}\Bigg[\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\text{SG}[\ell_{t,i}^{\text{clip}}]\ \log\pi_\theta(y_{i,t} \mid x, y_{i,<t}) + \beta\text{SG}[\rho_{i,t}]\mathcal{L}_{\text{OPD}}(x,\theta) + \alpha\text{SG}[\rho_{i,t}]f_{\mathcal{K}}(\pi_\theta, \pi_R; x, y_i)\Bigg]. \tag{C.2}$$

D Analysis on On-policy Context Distillation

Recent advancements like On-Policy Context Distillation (OPCD) (Ye et al., 2026) attempt to distill in-context knowledge c through on-policy KL matching. In our notation, the student distribution at prefix $s_t = (x, y_{<t})$ is

$$p_t(a) = \pi_\theta(a \mid x, y_{<t}),$$

and the privileged distribution is

$$q_t(a) = \pi_\theta(a \mid c, x, y_{<t}).$$

Let $\bar{q}_t = \text{SG}[q_t]$ and $\bar{p}_t = \text{SG}[p_t]$ denote the detached distributions at the current iterate, and set $\bar{D}_t = D_{\text{KL}}(\bar{p}_t \| \bar{q}_t)$. The reverse-KL full-vocabulary OPD objective used by SDPG is

$$\mathcal{L}_{\text{OPD},t} = D_{\text{KL}}(p_t \| \bar{q}_t) = \sum_{a\in\mathcal{V}} p_t(a)\log\frac{p_t(a)}{\bar{q}_t(a)}. \tag{D.1}$$

The corresponding negative student-side gradient at the fixed prefix is

$$-\nabla_\theta\mathcal{L}_{\text{OPD},t} = -\sum_{a\in\mathcal{V}} p_t(a)\left(\log\frac{p_t(a)}{\bar{q}_t(a)} + 1\right)\nabla_\theta\log p_t(a). \tag{D.2}$$

Equivalently, at the same fixed prefix and current iterate, the gradient can be written as an expectation over a detached student-token sample:

$$-\nabla_\theta\mathcal{L}_{\text{OPD},t} = \mathbb{E}_{a\sim\bar{p}_t}\left[\text{SG}\left[\bar{D}_t - \log\frac{\bar{p}_t(a)}{\bar{q}_t(a)}\right]\nabla_\theta\log p_t(a)\right],$$

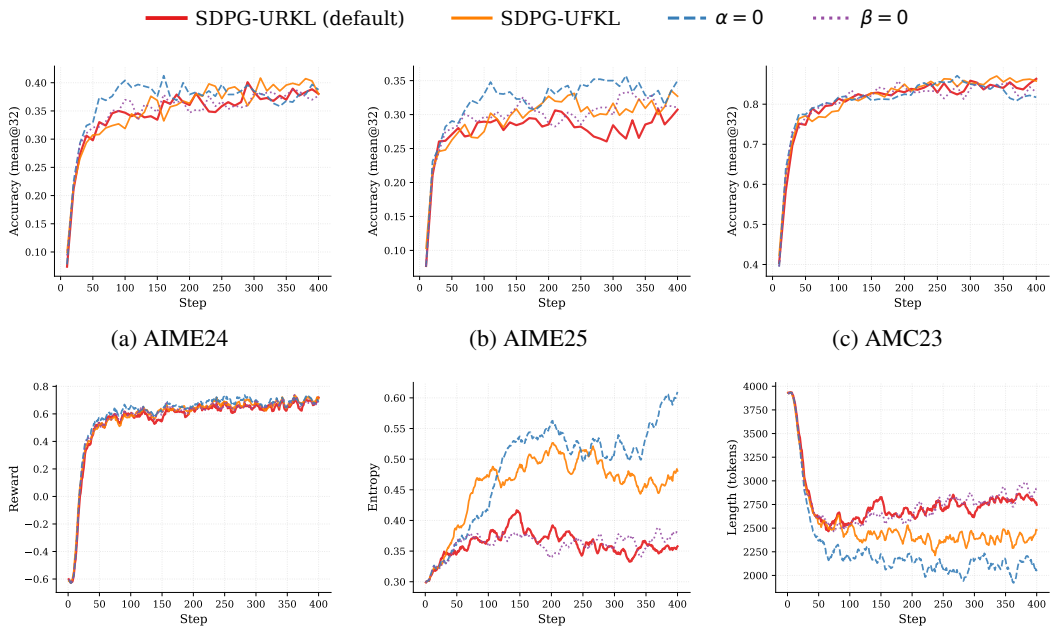
where the added baseline $1 + \bar{D}_t$ is valid because $\bar{p}_t = p_t$ at the iterate where the surrogate is formed. SDPG uses the explicit full-vocabulary objective $D_{\text{KL}}(p_t\|\bar{q}_t)$ in implementation, so its distillation component is exactly reverse-KL OPD, while the centered log-ratio expression clarifies the corresponding local policy-gradient signal.

However, pure self-distillation differs from SDPG in two important ways. First, it optimizes the privileged teacher signal without a binary outcome verifier, so it has no mechanism to prefer globally correct trajectories over locally plausible but incorrect ones. Second, it usually applies the teacher signal on all trajectories, including trajectories that the verifier would reject. SDPG addresses these issues by combining binary outcome rewards, positive-advantage gating, and reference-policy KL regularizationing.

E Ablation Studies

E.1 Effect of the Full-Vocabulary OPD Term and KL regularization

To isolate the individual contributions of the two non-outcome components in the SDPG objective (Eq. 3.1), we run two ablations on Qwen3-4B: setting $\alpha = 0$ removes the policy KL regularization, leaving only the binary outcome reward and full-vocabulary OPD term, and setting $\beta = 0$ removes the OPD term, recovering RPG without self-distillation.



(a) AIME24 (b) AIME25 (c) AMC23

(d) Reward (e) Entropy (f) Response Length

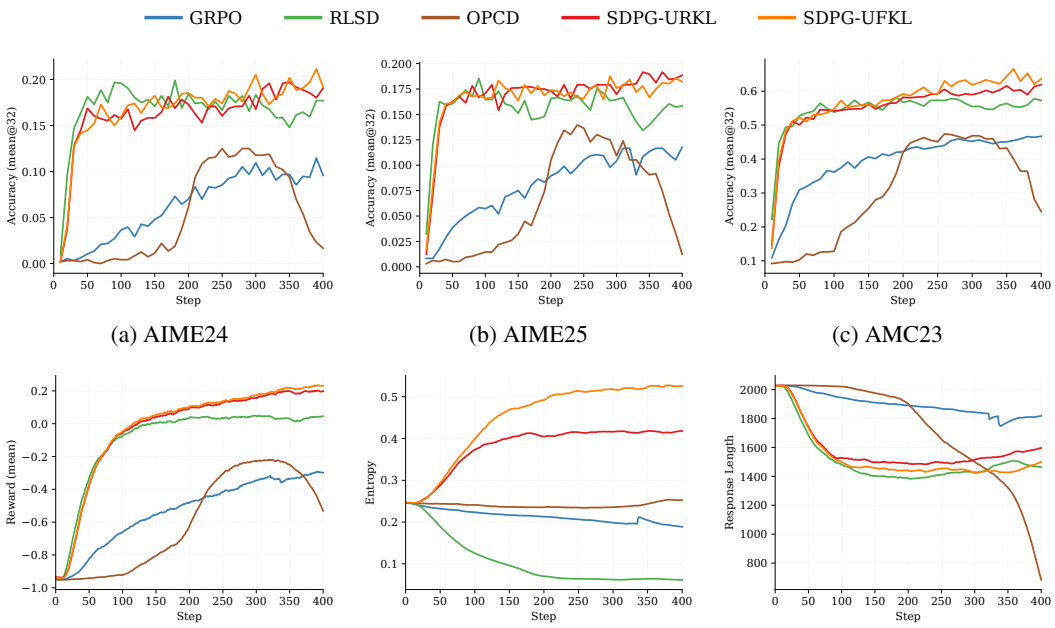
Figure 5: Ablation study on Qwen3-4B isolating the KL regularization (α) and the full-vocabulary OPD term (β). The top row are the result of (a) AIME24, (b) AIME25, (c) AMC23 (all pass@1 mean@32). The bottom row shows the (d) reward, (e) entropy, (f) response length. URKL (default) and UFKL are the full SDPG variants, while "$\alpha = 0$" removes the policy KL regularization and "$\beta = 0$" removes the OPD term.

The two ablations reveal complementary roles. Removing the full-vocabulary OPD term ($\beta = 0$) preserves the reward and length profiles of URKL but loses the early-training accuracy advantage on AIME24 and AIME25 (Figures 5a, b), confirming that privileged distillation is the primary driver of fast convergence on harder benchmarks. Removing the policy KL regularization ($\alpha = 0$), in contrast, achieves comparable or slightly higher accuracy on AIME24/25 but at the cost of severely shortened response length (around 2,000 tokens, Figure 5f) and rising entropy (Figure 5e), suggesting that without the anchor the student begins to deviate from coherent reasoning patterns. These observations indicate that full-vocabulary OPD provides dense supervision, while the KL regularization stabilizes policy updates, so that we choose to retain both components.

E.2 Robustness Across Model Scales: Qwen3-1.7B

To assess whether the design of SDPG generalizes beyond the 4B scale, we additionally run experiments on the smaller Qwen3-1.7B base model under the same training and evaluation protocol, with one additional baseline: OPCD (Ye et al., 2026). Results are shown in Figure 6 and Table 2.

The results on the 1.7B model corroborate the main findings at the 4B scale. SDPG-UFKL achieves the highest score on five of the six *Last*/*Best* columns, with SDPG-URKL taking the lead on AIME25, and the two SDPG variants together hold the top-two positions in every column except AIME24 *Best*. The accuracy gap over GRPO and RLSD opens within the first 50 steps and persists throughout



(a) AIME24 (b) AIME25 (c) AMC23

(d) Reward (e) Entropy (f) Response Length

Figure 6: Training dynamics and benchmark performance on Qwen3-1.7B trained with baseline algorithms and SDPG variants. The top row are the results of: (a) AIME24, (b) AIME25, (c) AMC23 (all pass@1 mean@32). The bottom row are the results of (d) reward, (e) entropy, (f) response length. SDPG-URKL and SDPG-UFKL outperform GRPO and RLSD across all three benchmarks. OPCD, a pure self-distillation baseline without reward, exhibits training instability after step 250, with accuracy on AIME and response length both collapsing.

Table 2: Performance on Qwen3-1.7B (pass@1, mean@32). *Last* is the score at step 400; *Best* is the peak across training. Column maximum in **bold**, second-best underlined.

	AIME24		AIME25		AMC23	
Method	Last	Best	Last	Best	Last	Best
GRPO	0.096	0.115	0.118	0.118	0.467	0.467
RLSD	0.177	0.199	0.158	0.185	0.573	0.582
OPCD	0.017	0.125	0.013	0.140	0.245	0.474
SDPG-URKL (ours)	0.191	0.197	**0.189**	**0.192**	0.620	0.620
SDPG-UFKL (ours)	**0.192**	**0.212**	0.182	0.188	**0.637**	**0.666**

training (Figures 6a–c). RLSD's entropy collapses below 0.1 by step 200 (Figure 6e), mirroring the pattern observed at the 4B scale, while SDPG-URKL and SDPG-UFKL maintain entropy above 0.4 throughout training. Notably, OPCD, as a pure self-distillation baseline, exhibits sharp degradation after step 250, where the AIME24 accuracy drops from 0.13 to 0.02, response length collapses to under 300 tokens, and reward turns sharply negative (Figures 6a, d, f). This instability supports the central design hypothesis of SDPG that full-vocabulary privileged distillation must be coupled with a binary outcome objective and a policy anchor to remain stable, especially on smaller models where pure imitation amplifies teacher imperfections.